OGD, NormExpGD

Lecturer: Kris Kitani *Scribes: Chih-Wei Wu, Yu-Jhe Li*

1 Review

In the last lecture, we have covered the Online Mirror Decent (OMD), Duality, and the analysis of OMD. We will review some significant points for each of these topics as follows:

1.1 Online Mirror Descent (OMD)

Before we directly go into OMD, recall we need to define some notation first and demonstrate how to generalize Follow the Regularized Leader (FTRL) w/ linear loss to OMD. To generalize FTRL linear loss sum, we should have the following notations:

1. $\boldsymbol{z}^{(1:t)} = \sum_{i=1}^{t} \boldsymbol{z}^{(i)}$ (sum of gradients)
2. $\boldsymbol{\theta} \triangleq -\boldsymbol{z}^{(1:t)}$ (iterating in the dual space)
3. $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \boldsymbol{z}^{(t)}$ (parameter of the dual space)

We can have the mirror linking function (from dual space θ to primal space w.):

$$\boldsymbol{w} = g(\boldsymbol{\theta})$$

The algorithm of Online Mirror Decent is presented in the Algorithm 1.

Algorithm 1 Online Mirror Decent (Convex set S, $g : \mathbb{R}^D \to S$)

1: **for** $t = 1, \cdots, T$ **do**
2:     RECEIVE $(\boldsymbol{f}^{(t)} : S \to R)$ ▷ Receive function
3:     $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \boldsymbol{z}^{(t)}$, $\boldsymbol{z} \in \partial f^{(t)}(\boldsymbol{w}^{(t)})$ ▷ Dual parameter update
4:     $\boldsymbol{w}_n^{(t+1)} \leftarrow g\left(\boldsymbol{\theta}^{(t+1)}\right)$ ▷ Mirror projection
5: **end for**

1.2 Duality

1.2.1 Convex Conjugate

Conjugate function is defined as:

$$\psi^*(\boldsymbol{\theta}) = \max_{\boldsymbol{w}} \left(\langle \boldsymbol{\theta}, \boldsymbol{w} \rangle - \psi(\boldsymbol{w}) \right)$$

The geometry of the conjugate can be seen as the intercept.

1.2.2 The property of convex conjugate

Derivative of the convex conjugate is:

$$\nabla_{\boldsymbol{\theta}} \psi^*(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \max_{\boldsymbol{w}} \left(\langle \boldsymbol{\theta}, \boldsymbol{w} \rangle - \psi(\boldsymbol{w}) \right) = \nabla_{\boldsymbol{\theta}} \left(\langle \boldsymbol{\theta}, \boldsymbol{w}^* \rangle - \psi(\boldsymbol{w}^*) \right) = \boldsymbol{w}^*$$

Slope the convex function is:

$$\nabla_{\boldsymbol{w}} \psi(\boldsymbol{w}) = \left. \frac{\partial \psi(\boldsymbol{w})}{\partial \boldsymbol{w}} \right|_{\boldsymbol{w}=\boldsymbol{w}^*} = \boldsymbol{\theta}$$

1.2.3 Fenchel-Young Inequality

Since we have the definition:

$$\psi^*(\boldsymbol{\theta}) = \max_{\boldsymbol{w}} \left(\langle \boldsymbol{\theta}, \boldsymbol{w} \rangle - \psi(\boldsymbol{w}) \right),$$

the Fenchel-Young Inequality can be defined as:

$$\psi^*(\boldsymbol{\theta}) \geq \left(\langle \boldsymbol{\theta}, \boldsymbol{w} \rangle - \psi(\boldsymbol{w}) \right)$$

1.2.4 Bregman Divergence

Bregman Divergence denotes the “distance” between two points according to some proximity function ψ, which can be defined as:

$$D_\psi(\boldsymbol{w} || \boldsymbol{u}) = \psi(\boldsymbol{w}) - \psi(\boldsymbol{u}) - \nabla \psi(\boldsymbol{u})^\top (\boldsymbol{w} - \boldsymbol{u}).$$

1.3 OMD Analysis

Now that we have the mathematical tools introduced in the previous section, we could derive the regret bound for Online Mirror Descent algorithm. The regret bound for Online Mirror Descent algorithm is:

$$\begin{aligned} R(\boldsymbol{u}) &= \sum_{t=1}^{T} \boldsymbol{w}^{(t)} \cdot \boldsymbol{z}^{(t)} - \boldsymbol{u} \cdot \boldsymbol{z}^{(t)} \\ &\leq \psi(\boldsymbol{u}) - \psi(\boldsymbol{w}^{(1)}) + \sum_{t=1}^{T} D_{\psi^*}(-\boldsymbol{z}^{(1:t)} || - \boldsymbol{z}^{(1:t-1)}) \end{aligned}$$

The first two terms $\psi(\boldsymbol{u}) - \psi(\boldsymbol{w}^{(1)})$ come from the **regularization function**, and the last term $\sum_{t=1}^{T} D_{\psi^*}(-\boldsymbol{z}^{(1:t)} || - \boldsymbol{z}^{(1:t-1)})$ comes from **Bregman Divergence under the convex conjugate of the regularization function**.

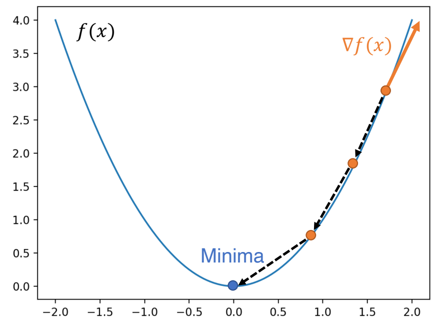

Figure 1: Geometric meaning of gradient descent.

2 Summary

2.1 Online Gradient Decent

2.1.1 Gradient Descent

Gradient descent is pretty much a standard approach for minimizing differentiable convex functions. In this lecture, we provide 3 perspectives to understand how gradient descent works.

Perspective 1: Geometric. The geometric intuition of gradient descent is illustrated in Figure 1. Given a convex and differentiable function $f : \mathbb{R}^N \to \mathbb{R}$, its gradient $\nabla f(\boldsymbol{w})$ at $\boldsymbol{w}$ could be calculated with $\nabla f(\boldsymbol{w}) = \{\frac{\partial f(\boldsymbol{w})}{w_1}, \cdots, \frac{\partial f(\boldsymbol{w})}{w_N}\}$. To find the minima of f, by moving step by step in the opposite direction of the gradient, we would eventually arrive at the minima of f. The complete algorithm of gradient descent is presented in Algorithm 2.

Algorithm 2 Gradient Decent (f)

1: $\boldsymbol{w}^{(0)} \leftarrow \mathbf{0}$
2: **for** $t = 1, \cdots, T$ **do**
3:     COMPUTE($\nabla f(\boldsymbol{w}^{(t-1)})$)
4:     $\boldsymbol{w}^{(t)} = \boldsymbol{w}^{(t-1)} - \eta \nabla f(\boldsymbol{w}^{(t-1)})$
5: **end for**

Perspective 2: Linear approximation with regularization. Mathematically, for a convex function $f : \mathbb{R}^N \to \mathbb{R}$, we could lower bound its value at $\boldsymbol{w}$ with Taylor series approximation at $\boldsymbol{u}$:

$$f(\boldsymbol{u}) \leq f(\boldsymbol{w}) + \langle \boldsymbol{u} - \boldsymbol{w}, \nabla f(\boldsymbol{w}) \rangle$$

The RHS serves as an approximation for $f(\boldsymbol{w})$, however, with a constraint. If we minimize the RHS, it would diverge to negative infinity instead of converging to the local minima of f. The reason is that the approximation is only accurate for values close to $\boldsymbol{w}$. To factor in this constraint, we should constrain the distance between $\boldsymbol{w}$ and $\boldsymbol{u}$ with squared L2 norm, i.e. $\min_w \|\boldsymbol{u} - \boldsymbol{w}\|_2^2$. Hence, to find the minima of function f, we should optimize the following objective function:

$$\min_{\boldsymbol{w}} \frac{1}{2} \|\boldsymbol{u} - \boldsymbol{w}\|_2^2 + \eta \Big(f(\boldsymbol{w}) + \langle \boldsymbol{u} - \boldsymbol{w}, \nabla f(\boldsymbol{w}) \rangle \Big)$$

And to find the w that minimizes the function f: (Note that we substitute $\boldsymbol{u}$ with $\boldsymbol{w}$, substitute $\boldsymbol{w}$ with $\boldsymbol{w}^{(t)}$)

$$\boldsymbol{w}^{(t+1)} = \arg\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}^{(t)}\|_2^2 + \eta\Big(f(\boldsymbol{w}^{(t)}) + \langle \boldsymbol{w} - \boldsymbol{w}^{(t)}, \nabla f(\boldsymbol{w}^{(t)})\rangle\Big)$$

The above equation is similar to the update step in Online Mirror Descent, where it optimizes a linear loss function with quadratic regularization.

Perspective 3: Isometric quadratic approximation. According Tayler expansion, we know that we could approximate a function f at $\boldsymbol{u}$ with a nearby location $\boldsymbol{w}$ by the following:

$$f(\boldsymbol{u}) \approx f(\boldsymbol{w}) + (\boldsymbol{u} - \boldsymbol{w})^T \nabla f(\boldsymbol{w}) + \frac{1}{2}(\boldsymbol{u} - \boldsymbol{w})^T \nabla^2 f(\boldsymbol{w})(\boldsymbol{u} - \boldsymbol{w})$$

For a function that is L-smooth, we could derive its upper bound function by the isometric quadratic approximation:

$$f(\boldsymbol{u}) \approx f(\boldsymbol{w}) + (\boldsymbol{u} - \boldsymbol{w})^T \nabla f(\boldsymbol{w}) + \frac{1}{2\eta}(\boldsymbol{u} - \boldsymbol{w})^T \boldsymbol{I}(\boldsymbol{u} - \boldsymbol{w}),$$

where η is a tunable variance parameter. For the proof that the above approximation is an upper bound of a L-smooth function, please refer to Appendix 3.1. To obtain the minima of f, one could optimize for the second approximated function:

$$\arg\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{w}\|^2 + \eta\Big(f(\boldsymbol{w}) + (\boldsymbol{u} - \boldsymbol{w})^T \nabla f(\boldsymbol{w})\Big)$$

By substituting $\boldsymbol{u}$ with $\boldsymbol{w}$, substituting $\boldsymbol{w}$ with $\boldsymbol{w}^{(t)}$, the formulation is same with **perspective 2**:

$$\boldsymbol{w}^{(t+1)} = \arg\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}^{(t)}\|_2^2 + \eta\Big(f(\boldsymbol{w}^{(t)}) + \langle \boldsymbol{w} - \boldsymbol{w}^{(t)}, \nabla f(\boldsymbol{w}^{(t)})\rangle\Big)$$

To obtain the minima, we use the simple method of deriving its gradient and find the solution of the gradient:

$$\boldsymbol{w} = \boldsymbol{w}^{(t)} - \eta \nabla f(\boldsymbol{w}^{(t)})$$

This displays resemblance to the update rule of the Weighted Majority Algorithm, Online Perceptron Algorithm, and the Follow The Regularized Leader Algorithm. In short, the final equation is the solution to the gradient descent on the function f from a **quadratic approximation** of the loss function f.

2.1.2 Stochastic Gradient Descent

Although gradient descent is guaranteed to reach the local minimum mathematically, its computation speed is not desirable in practice. The problem is that computing gradient of the function ∇f over all training example is expensive sometimes. An alternative is to perform the **Stochastic Gradient Descent**, where instead of updating parameter after computing gradient of all examples, it updates the parameter after computing each example's gradient. An overview of the stochastic

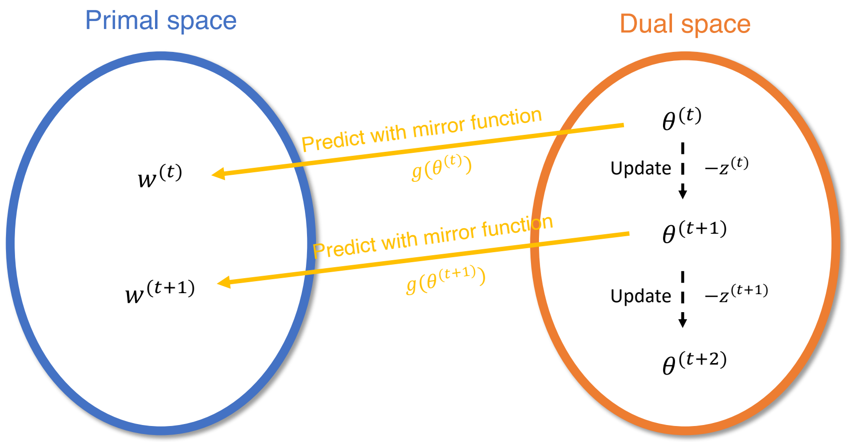

Figure 2: Dual space of the Online Mirror Descent.

gradient descent is demonstrated in Algorithm 3. In comparison to the vanilla gradient descent, computing the gradient of single sample or part of the full data is much faster. Although the direction of each iteration would not always point at the descending direction, the expected value of the direction would equal the gradient direction. Consequently, stochastic gradient descent has similar convergence bound as gradient descent.

Algorithm 3 Stochastic Gradient Decent (f)

```
1: w^(1) ← 0
2: η > 0
3: for t = 1, ··· , T do
4:     z ∼ D                         ▷ Sample single sample from data distribution
5:     v^(t) = ∇f_z(w^(t−1))         ▷ Fast to compute
6:     w^(t) = w^(t−1) − η v^(t)
7: end for
```

2.1.3 Online (Projected Sub-) Gradient Descent as OMD

In this subsection, we are going to show that Online Gradient Descent is a special case of Online Mirror Descent (OMD). Recall that the Online Mirror Descent updates its primal parameter with the dual parameter by a mirror function $g(\boldsymbol{\theta})$, as shown in Figure 2. The mirror function, or aliased with weight prediction rule, is written as:

$$\boldsymbol{w}^{(t+1)} = \arg\min_{\boldsymbol{w}} \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \psi(\boldsymbol{w})$$

Online Gradient Descent is just a special case of Online Mirror Descent, by defining this **quadratic regularization function**: $\phi(\boldsymbol{w}) = \frac{1}{2\eta}\|\boldsymbol{w}\|_2^2$, and defining a **linear loss function**: $f(\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{\theta} \rangle$. Namely, the weight prediction rule for Online Gradient Descent is written as:

$$\boldsymbol{w}^{(t+1)} = \arg\min_{\boldsymbol{w}} \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \frac{1}{2\eta}\|\boldsymbol{w}\|_2^2$$

Furthermore, we could derive the minimization function by finding the solution of its gradient:

$$\begin{aligned}\mathcal{L} &= \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \frac{1}{2\eta}\sum_n w_n^2 \\ \frac{\partial \mathcal{L}}{\partial w_n} &= \theta_n + \frac{1}{2\eta}2w_n = 0 \\ &\Rightarrow w_n = -\eta\theta_n\end{aligned}$$

The last equation actually projects the parameter in the dual space back into the parameter in the primal space. Therefore, we conclude the mirror function for Online Gradient Descent is:

$$g(\boldsymbol{\theta}) = -\eta\boldsymbol{\theta}$$

There is actually 2 variants of the Online Gradient Descent mirror function. The first one uses the above weight prediction rule under the assumption that the sub-gradient of the function is constraint within certain range. This is called the Online Sub-Gradient Descent, as depicted in Algorithm 4. The second one adds a convex set projection function to the weight prediction rule: $g(\boldsymbol{\theta}) = \prod_{\theta \to S}(-\eta\boldsymbol{\theta})$. This is called the Online Projected Sub-Gradient Descent, as displayed in Algorithm 5. Both of them falls into the family of Online Gradient Descent, which we that it is a special case of Online Mirror Descent with linear loss and quadratic regularizer.

Algorithm 4 Online Sub-Gradient Decent (η)

1: **for** $t = 1, \cdots, T$ **do**
2: $\quad \boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \boldsymbol{z}^{(t)},\ \boldsymbol{z} \in \partial f^{(t)}(\boldsymbol{w}^{(t)})$ ▷ Dual parameter update
3: $\quad \boldsymbol{w}_n^{(t+1)} = -\eta\boldsymbol{\theta}$ ▷ Mirror projection
4: **end for**

Algorithm 5 Online Projected Sub-Gradient Decent (η)

1: **for** $t = 1, \cdots, T$ **do**
2: $\quad \boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \boldsymbol{z}^{(t)},\ \boldsymbol{z} \in \partial f^{(t)}(\boldsymbol{w}^{(t)})$ ▷ Dual parameter update
3: $\quad \boldsymbol{w}_n^{(t+1)} = \prod_{\theta \to S} -\eta\boldsymbol{\theta}$ ▷ Mirror projection
4: **end for**

2.1.4 Online Gradient Descent Analysis

Now we would like to derive a regret bound as:

$$R_{OGD} \leq DG\sqrt{T}$$

where $D = \max \|\boldsymbol{u}\|_2, \boldsymbol{u} \in S$ (assumption on the magnitude of the primal parameter) and $G = \max \|\boldsymbol{z}\|_2, \boldsymbol{z} \in \partial f(\boldsymbol{w})$ (assumption on the magnitude of sub-gradient). Recall the general regret bound as:

$$R(\boldsymbol{u}) = \sum_{t=1}^{T} \langle \boldsymbol{w}^{(t)}, \boldsymbol{z}^{(t)} \rangle - \langle \boldsymbol{u}, \boldsymbol{z}^{(1:T)} \rangle \leq \psi(\boldsymbol{u}) - \psi(\boldsymbol{w}^{(1)}) + \sum_{t=1}^{T} D_{\psi^*}(-\boldsymbol{z}^{(1:t)} || - \boldsymbol{z}^{(1:t-1)})$$

We can start from the regret bound of OMD to derive the regret bound of OGD:

$$
\begin{aligned}
R(\boldsymbol{u}) \leq & \psi(\boldsymbol{u}) - \psi(\boldsymbol{w}^{(1)}) + \sum_{t=1}^{T} D_{\psi^*}(\boldsymbol{\theta}^{(t+1)}||\boldsymbol{\theta}^{(t)}) \\
= & \psi(\boldsymbol{u}) - \psi(\boldsymbol{w}^{(1)}) + \sum_{t=1}^{T} \psi^*(\boldsymbol{\theta}^{(t-1)}) - \psi^*(\boldsymbol{\theta}^{(t)}) - \nabla\psi^*(\boldsymbol{\theta}^{(t-1)})(\boldsymbol{\theta}^{(t)})(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}) \\
= & \frac{1}{2\eta}\|\boldsymbol{u}\|_2^2 - \frac{1}{2\eta}\|\boldsymbol{w}\|_2^2 + \sum_{t=1}^{T} \frac{1}{2\eta}\|\boldsymbol{\theta}^{(t+1)}\|_2^2 - \|\boldsymbol{\theta}^{(t)}\|_2^2 - \frac{1}{\eta}\boldsymbol{\theta}^{(t)}(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}) \\
= & \frac{1}{2\eta}\|\boldsymbol{u}\|_2^2 - \frac{1}{2\eta}\|\boldsymbol{w}\|_2^2 + \sum_{t=1}^{T} \frac{1}{2\eta}\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\|_2^2 \\
= & \frac{1}{2\eta}\|\boldsymbol{u}\|_2^2 - \frac{1}{2\eta}\|\boldsymbol{w}\|_2^2 + \sum_{t=1}^{T} \frac{1}{2\eta}\|(-\boldsymbol{z}^{(1:t)}) - (-\boldsymbol{z}^{(1:t-1)})\|_2^2 \\
\leq & \frac{1}{2\eta}||\boldsymbol{u}||_2^2 + \sum_{t=1}^{T} \frac{1}{2\eta}||\boldsymbol{z}^{(t)}||_2^2
\end{aligned}
$$

Recall:

$$
\begin{aligned}
D &= \max ||\boldsymbol{u}||_2 \quad \boldsymbol{u} \in S \\
G &= \max ||\boldsymbol{z}||_2 \quad \boldsymbol{z} \in \partial f(\boldsymbol{w})
\end{aligned}
$$

So we can derive:

$$
R_{ODG} \leq \frac{D^2}{2\eta} + \frac{\eta}{2}T^2 G
$$

To find the optimal η, we take the derivative:

$$
\frac{d}{d\eta}\left\{\frac{1}{\eta}D^2 + \frac{\eta}{2}G^2 T\right\} = 0 \rightarrow \eta = \frac{D}{G\sqrt{T}}
$$

Then we can subscribe back to the regret:

$$
\begin{aligned}
R_{ODG}(\boldsymbol{u}) \leq & \frac{D^2}{2\eta} + \frac{\eta}{2}T^2 G \\
= & DG\sqrt{T}
\end{aligned}
$$

2.2 Online Normalized Exponentiated Gradient Descent

Recall the Online Gradient Decent in the Algorithm 1, we can actually have the Norm-Exponentiated-Gradient summarized in the Algorithm 6. We will show how we can derive the Normalized Exponentiated Gradient from Algorithm 1 to Algorithm 6 in some steps below.

Algorithm 6 Norm-Exponentiated-Gradient (η)

for $t = 1, \cdots, T$ **do**
 $\boldsymbol{z} \in \partial f^{(t)}(\boldsymbol{w}^{(t)})$
 $\boldsymbol{w}^{(t+1)} \propto \boldsymbol{w}^{(t)} \exp(\eta \boldsymbol{z}^{(t)})$
end for

In the first place, we will define the regularization function $\psi(w)$ as:

$$\psi(\boldsymbol{w}) = \sum_{k=1}^{K} \boldsymbol{w}_k \log \boldsymbol{w}_k \quad \boldsymbol{w} \in \mathbb{S}^K. \tag{1}$$

You may see equation 1 as negative entropy and K-simplex constraint. Then when we define the loss as:

$$f(\boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{\theta} \rangle,$$

the prediction rule will become:

$$\begin{aligned} \boldsymbol{w}^{(t+1)} &= \underset{\boldsymbol{w}}{\operatorname{argmin}} \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \psi(\boldsymbol{w}) \\ &= \underset{\boldsymbol{w} \in \mathbb{S}^K}{\operatorname{argmin}} \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \sum_{k=1}^{K} \boldsymbol{w}_k \log \boldsymbol{w}_k \end{aligned} \tag{2}$$

Now we can then add simplex constraint to the objective in the equation 2 as:

$$\boldsymbol{w}^{(t+1)} = \underset{\boldsymbol{w} \in \mathbb{S}^K}{\operatorname{argmin}} \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \sum_{k=1}^{K} \boldsymbol{w}_k \log \boldsymbol{w}_k + \lambda \left(1 - \sum_k \boldsymbol{w}_k \right) \tag{3}$$

The Lagrangian can be derived from the equation as:

$$\mathcal{L} = \langle \boldsymbol{w}, -\boldsymbol{\theta}^{(t+1)} \rangle + \frac{1}{\eta} \sum_{k=1}^{K} \boldsymbol{w}_k \log \boldsymbol{w}_k + \lambda \left(1 - \sum_k \boldsymbol{w}_k \right)$$

Now we would like to solve the minimum of the Lagrangian equation. We take the partial derivative w.r.t. $\boldsymbol{w}_n$ as:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_n} &= -\boldsymbol{\theta}_n + \frac{1}{\eta}(1 + \log \boldsymbol{w}_n) - \lambda \\ \frac{1}{\eta} \log \boldsymbol{w}_n &= \boldsymbol{\theta}_n - \frac{1}{\eta} + \lambda \\ \boldsymbol{w}_n &= \exp(\eta \boldsymbol{\theta}_n - (1 - \eta\lambda)) \\ \boldsymbol{w}_n &= \frac{\exp(\eta \boldsymbol{\theta}_k)}{\exp(1 - \eta\lambda)} \end{aligned}$$

We can then summarize the linear loss plus the entropic regularization as:

- Minimizer for linear loss and entropic regularization:

$$w_n = \frac{\exp(\eta \theta_k)}{\exp(1 - \eta\lambda)}$$

- Dual parameter update:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \boldsymbol{z}^{(t)}, \quad \boldsymbol{z}^{(t)} \in \partial f^{(t)}(\boldsymbol{w}^{(t)})$$

- Mirror function (enforces the geometry of the problem, e.g. probability simplex):

$$g(\boldsymbol{\theta}) = \frac{\exp(\eta\boldsymbol{\theta})}{\sum_{n'} \exp(\eta\theta_{n'})}$$

Since now we have the rule for dual parameter update and the mirror function, Online Normalized Exponentiated Gradient Descent then can the be arranged in the Algorithm 7.

Algorithm 7 Online Norm-Exp-GD (η)

for $t = 1, \cdots, T$ **do**
  $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \boldsymbol{z}^{(t)}, \boldsymbol{z}^{(t)} \in \partial f^{(t)}(\boldsymbol{w}^{(t)})$ ▷ Dual parameter update
  $\boldsymbol{w}^{(t+1)} \propto \exp\left(\eta \boldsymbol{\theta}^{(t+1)}\right)$ ▷ Mirror projection
end for

Let's make the connection to ONEGD more clear as follows:

$$\begin{aligned}
\boldsymbol{w}_n^{(t+1)} =& \frac{\exp\left(\eta\boldsymbol{\theta}_n^{(t+1)}\right)}{\sum_{n'}\exp(\eta\boldsymbol{\theta}_{n'})} \\
=& \frac{\exp\left(\eta(\boldsymbol{\theta}_n^{(t)} - \boldsymbol{z}_n^{(t)})\right)}{\sum_k \exp\left(\eta(\boldsymbol{\theta}_k^{(t)} - \boldsymbol{z}_k^{(t)})\right)} \\
=& \exp(\eta\boldsymbol{\theta}_n^{(t)}) \frac{\exp(-\eta \boldsymbol{z}_n^{(t)})}{\sum_k \exp(\eta\boldsymbol{\theta}_k^{(t)})\exp(-\eta\boldsymbol{z}_k^{(t)})} \cdot \frac{\sum_j \exp(\eta\boldsymbol{\theta}_j^{(t)})}{\sum_j \exp(\eta\boldsymbol{\theta}_j^{(t)})} \\
=& \frac{\boldsymbol{w}_n^{(t)}\exp(-\eta\boldsymbol{z}_n^{(t)})}{\sum_k \boldsymbol{w}_k^{(t)}\exp(-\eta\boldsymbol{z}_k^{(t)})}
\end{aligned} \tag{4}$$

From equation 4, we can see it exhibits the same update as the weighted majority algorithm without normalizing:

$$\boldsymbol{w}_n^{(t+1)} \propto \boldsymbol{w}_n^{(t)} \exp(-\eta\boldsymbol{z}_n^{(t)}).$$

Let's review the Hedge algorithm which is presented in Algorithm 8.

Algorithm 8 Hedge algorithm

$\mathbf{w}^{(1)} \leftarrow \{\boldsymbol{w}_n^{(1)} = 1\}_{n=1}^N$ ▷ Weight initialization
for $t = 1, \cdots, T$ **do**
  Receive $(\mathbf{x}^{(t)} \in \{-1, 1\}^N)$ ▷ Receive experts predictions
  $I \sim$ Multinomial$(\mathbf{w}^{(t)}/\Phi^{(t)})$, where $\Phi^{(t)} = \sum_{n=1}^N w_n^{(t)}$
  $\hat{y}^{(t)} = h_i(\mathbf{x}^{(t)})$ ▷ Make learner prediction via sampling
  Receive $(y^{(t)} \in \{-1, 1\})$ ▷ Receive actual answer
  $\boldsymbol{w}_n^{(t+1)} = \boldsymbol{w}_n^{(t)} e^{-\beta\cdot\mathbf{1}[y^{(t)} \neq h_n(\mathbf{x})^{(t)}]}$ ▷ Weight update
end for

You can see this is actually the unnormalized exponentiated gradient descent, which comes from entropic regularization.

References

[1] D. P. Bertsekas. *Nonlinear Programming 2nd Edition.*

[2] Wikipedia. Lipschitz continuity.

3 Appendix

3.1 Lipschitz continuity

A Lipschitz continuous function f is a function that is limited how fast it can change by a Lipschitz constant L [2]. The Lipschitz constant L represents the absolute value of the slope of 2 lines. These 2 lines when sliding along the function f itself, would never touch the function f itself. The concept is illustrated in Figure 3. Formally, a function $f(\cdot)$ is called a L-Lipschitz continuous function over

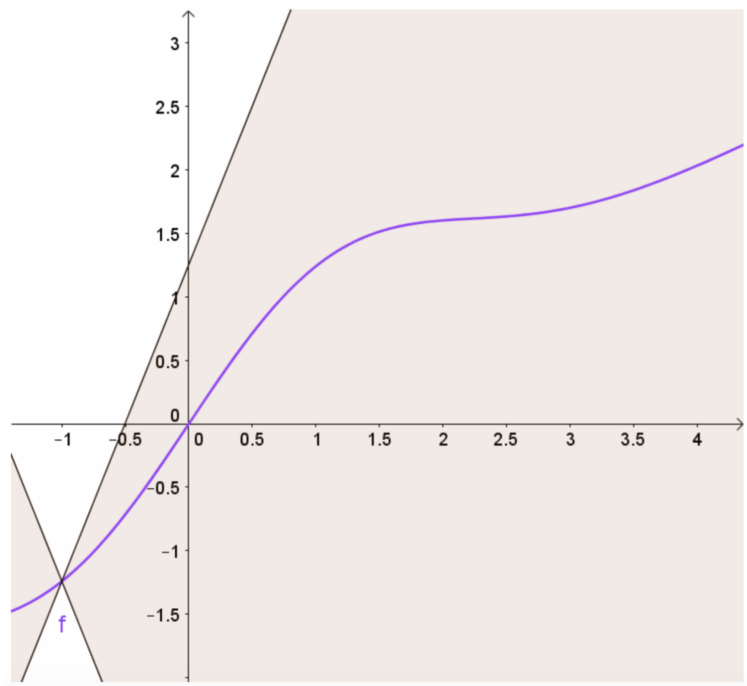

Figure 3: Concept of a Lipschitz continuous function. (Image from Wikipedia [2])

a set S with respect to a metric $\|\cdot\|$ if for all $\boldsymbol{u}, \boldsymbol{w} \in S$:

$$|f(\boldsymbol{u}) - f(\boldsymbol{w})| \leq L\|\boldsymbol{u} - \boldsymbol{w}\|$$

With this property, we can proof that the function f is upper bounded by:

$$f(\boldsymbol{u}) \leq f(\boldsymbol{w}) + (\boldsymbol{u} - \boldsymbol{w})^T \nabla f(\boldsymbol{w}) + \frac{L}{2}\|\boldsymbol{u} - \boldsymbol{w}\|_2^2$$

Here, we rephrase the above property as the following lemma [1].

Lemma 1 (Descent Lemma). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continously differentiable, and let x and y be two vectors in R^n. Suppose that*

$$\|\nabla f(x + ty) - \nabla f(x)\| \leq Lt\|y\|, \quad \forall t \in [0, 1],$$

where L is some scalar. Then

$$f(x + y) \leq f(x) + y'\nabla f(x) + \frac{L}{2}\|y\|^2$$

Proof. Let t be a scalar parameter and let $g(t) = f(x + ty)$. The chain rule yields $(dg/dt)(t) = y'\nabla f(x + ty)$. Now

$$\begin{aligned}
f(x+y) - f(x) = g(1) - g(0) &= \int_0^1 \frac{dg}{dt}(t)dt = \int_0^1 y'\nabla f(x + ty)dt \\
&\leq \int_0^1 y'\nabla f(x)dt + \left| \int_0^1 y'(\nabla f(x + ty) - \nabla f(x))dt \right| \\
&\leq \int_0^1 y'\nabla f(x)dt + \int_0^1 \|y\| \cdot \|\nabla f(x + ty) - \nabla f(x)\|dt \\
&\leq y'\nabla f(x) + \|y\| \int_0^1 Lt\|y\|dt \\
&= y'\nabla f(x) + \frac{L}{2}\|y\|^2.
\end{aligned}$$

$\square$